

## Expt:- Implement a Yolo Model to detect objects.

(Goal & Objectives) :-

• Python and PyTorch C.

• Model API based C.

• Train model & store C.

• Detect model + visual + visual = handwritten -

• Training (PyTorch) API -

• Accuracy & few steps higher -

• Check API visual C.

• Visual API, Island spotting

Pseudocode:-

- 1.) Import torch, torchvision, matplotlib, Sklearn.
- 2.) Load Pascal VOC dataset using torchvision, datasets, VOC detection.
- 3.) Reproces images  $\rightarrow$  resize (448x448), normalization.
- 4.) Build Yolo model  $\rightarrow$  CNN backbone.
- 5.) Define Yolo loss.
- 6.) Train the model for N epochs, track loss.
- 7.) Plot training loss and map.
- 8.) Print final classification report -- .

• check test no Island spotting -

• Visual API -

(Aim) :-

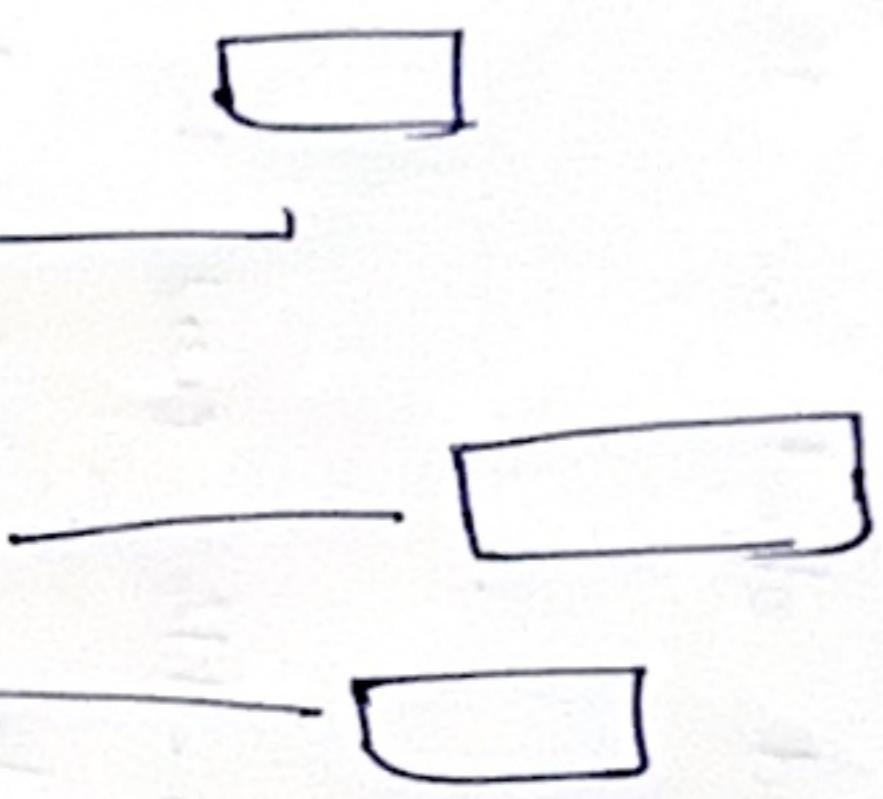
To implement a simplified Yolo-style object and using Pytorch to detect object on the common dataset objects.

Objectives:-

- 1.) Build a tiny-Yolo-like CNN in Pytorch.
- 2.) Train and evaluate on Pascal dataset.
- 3.) Generate classification report and graph for loss and map.
- 4.) Observe detection performance and analyze result.

### Observation :-

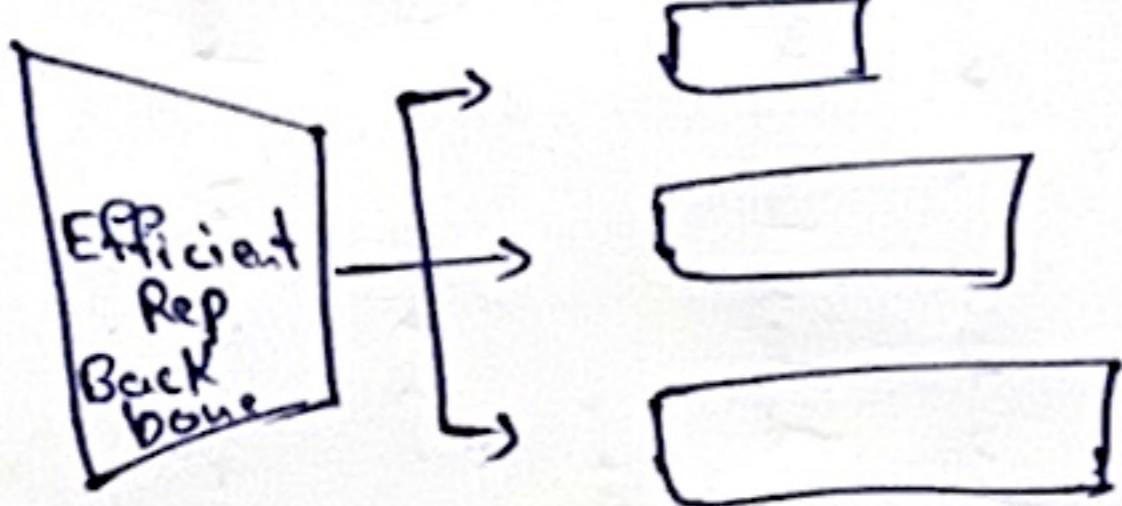
- The training loss should gradually decrease across epochs showing that the YOLO model is learning basic localization.



Rep Pow.

### YOLO model

Input



### Result:-

The model successfully runs, loss curve showing decreasing loss values. Provided evaluation matrix and visualization.

Sop. J